

The School Bus Routing Problem: An Analysis and Algorithm

R. Lewis¹, K. Smith-Miles², and K. Phillips³

¹ School of Mathematics, Cardiff University, Wales.

² School of Mathematical Sciences, Monash University, Australia.

³ Visible Services and Transport, Vale of Glamorgan Council, Wales.

LewisR9@cf.ac.uk, kate.smith-miles@monash.edu,

kwphillips@valeofglamorgan.gov.uk

Abstract. In this paper we analyse a flexible real world-based model for designing school bus transit systems and note a number of parallels between this and other well-known combinatorial optimisation problems including the vehicle routing problem, the set covering problem, and one-dimensional bin packing. We then describe an iterated local search algorithm for this problem and demonstrate the sort of solutions that we can expect with different types of problem instance.

1 Introduction

Vehicle routing problems (VRPs) involve identifying routes for a fleet of vehicles that are to serve a set of customers. Often they are expressed using an edge-weighted directed graph $G = (V, E)$, where the vertex set $V = \{v_0, v_1, \dots, v_n\}$ represents a single depot and n customers (v_0 and v_1, \dots, v_n respectively), and the weighting function $w(u, v)$ gives the travel distance (or travel time) between each pair of vertices $u, v \in V$.

Since the work of Dantzig and Ramser in the late 1950s [4], a multitude of VRP formulations have been considered in the literature [7]. These include using time-windows for visiting certain customers, placing limitations on the lengths of individual routes, the partitioning of customers into pick-up and delivery locations, and the dynamic recalibration of routes subject to the arrival of new customer requests during the transportation period [11].

Solutions to most VRP problems can be expressed by a set of routes $\mathcal{R} = \{R_1, \dots, R_k\}$ using one vehicle per-route. In the *classical* VRP, each route should be a simple cycle in G such that:

$$R_i \cap R_j = \{v_0\} \quad \forall R_i, R_j \in \mathcal{R} \quad (1)$$

$$\bigcup_{i=1}^k R_i = V \quad (2)$$

These constraints specify that each customer should be assigned to exactly one route, and that all routes should start and end at the depot v_0 . A variation on

this is the *open* VRP in which, instead of cycles, all routes must be simple paths containing v_0 as one terminal vertex, meaning that routes either start or end at the depot, but not both [8].

In the *time constrained* VRP, extra realism is added by specifying that the total weight of edges in each route should be less than a given maximum—e.g., to ensure that driving time regulations are obeyed. In the *capacitated* VRP, meanwhile, maximum capacities are specified for each vehicle, and weights are also added to the vertices v_1, \dots, v_n in G . These vertex weights represent the size of the items being delivered to each customer, and we require the total size of items delivered by each vehicle to not exceed its maximum capacity.

The *split-delivery* VRP extends the capacitated VRP by relaxing Constraint (1) to simply: $v_0 \in R, \forall R \in \mathcal{R}$. This allows more than one vehicle to visit a customer and therefore permits a delivery to be made in many parts. Unlike the capacitated VRP, this relaxation also allows the minimum number of routes/vehicles in a solution to meet the lower bound of $\lceil (\sum_{i=1}^n w(v_i)) / C \rceil$, where $w(v)$ gives the weight of a vertex and C is the maximum capacity of the vehicles [15].

Objective functions for the VRP can depend on many real world factors. Most commonly we seek to minimise the number of vehicles used, the total length of the routes, or some combination of the two. In other cases we might also be concerned with the waiting times of customers, the obeying of time windows, avoiding traffic jams, or meeting individual drivers' needs. A useful survey presenting a taxonomy of the various types of VRP can be found in [5].

In this paper we look at the problem of arranging school bus transport. This problem is often cited as a type of VRP applicable in the real-world, though historically it has been less studied than other variants. One reason for this is that school transport solutions usually only involve visiting a subset of the available stopping points (bus stops); hence the issue of choosing *which* bus stops to visit adds an extra layer of complexity to the problem. Indeed, Park et al. [10] note that bus stop selection is often omitted in the VRP literature altogether. One notable exception to this is due to Schittekat et al. [14], who use a problem based on the requirements of the Belgian school system; however, their formulation involves assumptions not considered here, most notably their limitation that bus stops can only be visited by a maximum of one vehicle in a solution.

The problem considered here is rather generic and was originally supplied by the third author of this paper, whose organisation is responsible for arranging school transport in the south of Wales (population 2.2m). Like many countries, school transport in Wales is organised by local government and then run by private bus companies. A few months before the start of the school year, a list of addresses is compiled containing all school children eligible for school transport (usually those who are in the school's catchment area but not within a reasonable walking distance). Each school is then considered individually, and a set of suitable bus routes are drawn up to serve all qualifying students. These routes are then put out for public tender, with bus companies bidding for the contracts. A yearly contract for a 70-seat bus typically ranges from GBP £25,000 to £35,000,

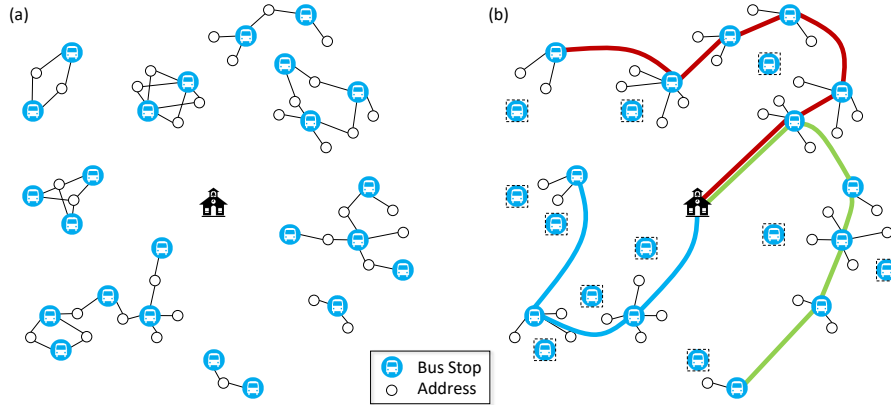


Fig. 1. (a) Example problem instance; (b) Example solution using $k = 3$ routes. Bus stops with dotted outlines are not used (i.e., are not members of V_1').

though these costs can increase further for longer journeys and for routes requiring a chaperone (i.e., routes with young children). It is therefore critical to try to reduce the number of buses used by each school. Note, however, that government guidelines also specify that journeys should not be too lengthy (less than 45 minutes for under-11s, and one hour for under-18s), though exceptions can be made for schools with very large catchment areas.

2 Problem Definition

The school bus routing problem (SBRP) considered here can be more formally stated using two sets of vertices. The first vertex set V_1 contains one school, v_0 , and n bus stops v_1, \dots, v_n . An edge set E_1 then contains directed edges between each $u, v \in V_1$ in each direction, making the graph (V_1, E_1) a complete digraph. A nonnegative weighting function $t(u, v)$ is also used to define the shortest driving time between each vertex pair.

The second vertex set V_2 defines the set of student *addresses*, with the weight $s(v) \in \mathbb{Z}^+$ of each $v \in V_2$ giving the number of students at this address requiring school transport. As part of the problem, a parameter m_w is defined stating the maximum distance that students are expected to walk from their home address to a bus stop. A second set of edges is thus used to signify bus stops within walking distance of each address: $E_2 = \{\{u, v\} : u \in V_2 \wedge v \in V_1 \wedge w(u, v) \leq m_w\}$, where $w(u, v)$ gives the shortest walking distance between each $u \in V_2$ and $v \in V_1$. Also, students living within m_e distance units of their school are not considered eligible for school transport; consequently, $w(u, v_0) \geq m_e \forall u \in V_2$.

The graph $(V_1 - \{v_0\}, V_2, E_2)$ therefore constitutes an undirected bipartite graph with potentially many components, as illustrated in Fig. 1(a). Note that if there exists an address $u \in V_2$ with just one incident edge $\{u, v\} \in E_2$, then the bus stop $v \in V_1$ is *compulsory*, since it must be included in a solution in order

to satisfy the needs of address u . We can also assume that $(V_1 - \{v_0\}, V_2, E_2)$ contains no isolated vertices: such vertices in $(V_1 - \{v_0\})$ would give a bus stop with no address within walking distance and can therefore be removed from the problem; isolated vertices in V_2 define an address with no suitable bus stop, making the problem unsolvable (in practice, an additional bus stop would need to be added to serve such an address).

A *feasible* solution to the SBRP is a set of routes $\mathcal{R} = \{R_1, \dots, R_k\}$ in which each route $R \in \mathcal{R}$ is a simple path served by a single bus of capacity m_c . Each bus then travels to the school v_0 after visiting the terminal vertex on its path. The following constraints need to be satisfied.

$$\bigcup_{i=1}^k R_i = V'_1 \quad (3)$$

$$\forall u \in V_2 \quad \exists v \in V'_1 : \{u, v\} \in E_2 \quad (4)$$

$$s(R) \leq m_c \quad \forall R \in \mathcal{R} \quad (5)$$

$$t(R) \leq m_t \quad \forall R \in \mathcal{R} \quad (6)$$

Here, V'_1 is a subset of $(V_1 - \{v_0\})$ that should satisfy Constraint (4): that is, for each address $u \in V_2$, the set V'_1 should contain at least one bus stop within walking distance. Constraint (5) then specifies that the total number of students boarding the bus on a route R , denoted by $s(R)$, does not exceed the maximum bus capacity m_c . Similarly, Constraint (6) states that the total journey time $t(R)$ of each route should not exceed the stated time limit m_t . The aim is to then produce a feasible solution that minimises the number of routes k . An example solution to this problem is shown in Fig. 1(b). Note that these constraints allow bus stops to be included in more than one route, as is the case in the diagram. We call these bus stops *multistops*, their presence allowing different bus routes to split and merge as needed.

In our algorithm, our strategy is to relax Constraint (6) while ensuring that (3)–(5) are always satisfied. In doing so, a number of assumptions are made. First, students are always assigned to the bus stop in V'_1 closest to their home. Second, all students are given bus passes that only allow them to travel on one particular route. This avoids situations where too many students might board a bus at a multistop, thereby making it too full to serve students at a later non-multistop. Third, solutions only concern buses travelling *to* school. After-school routes are assumed to follow the same paths in reverse, with any discrepancies in travel time due to one-way streets, etc. not being considered.

Our final assumption involves the use *dwell times* within a journey. These measure the time spent servicing each bus stop, including decelerating, opening doors, loading passengers, and rejoining the traffic stream. Dwell times are influenced by many factors including the number of boarding passengers, the size and position of the doors, the age of the passengers, and traffic density. Commonly, simple linear models $y = a + bx$ are used to estimate a dwell time y , where x gives the number of boarding passengers, b gives the boarding time per-passenger, and a captures all remaining delays. We follow this approach here:

Definition 1. The journey time $t(R)$ of a route $R = (u_1, u_2, \dots, u_l) \in \mathcal{R}$ is calculated,

$$t(R) = \left(\sum_{i=1}^{l-1} t(u_i, u_{i+1}) \right) + t(u_l, v_0) + \left(\sum_{i=1}^l a + b \cdot s(u_i, R) \right), \quad (7)$$

where $s(u_i, R)$ denotes the number of students boarding the bus on route R at bus stop u_i .

In our case we use the values $a = 15$ and $b = 5$ (seconds), which are consistent with those recommended in [2, 12, 16].

3 Problem Analysis

In this section we now make some observations about the complexity of the SBRP and its underlying subproblems.

Theorem 1. The task of finding a feasible solution with a minimum number of routes is NP-hard.

Proof. Let (V_1, V_2, E_2) be a graph such that $\deg(u) = 1 \forall u \in V_2$. This means that, for all bus stops $v \in (V_1 - \{v_0\})$, (a) v is compulsory and must appear in at least one route, and (b) the number of boarding students is fixed at $\sum_{u \in \Gamma(v)} s(u)$. This also implies the dwell times at each bus stop are fixed. This special case is equivalent to the NP-hard time-constrained capacitated split-delivery VRP, itself a generalisation of the NP-hard time-constrained VRP.

A similar proof of NP-hardness considers a generalisation of the above in which each component in $(V_1 - \{v_0\}, V_2, E_2)$ is a complete bipartite graph. In this case, all students can be assigned to a bus by including in V_1' exactly one bus stop from each component, making the problem a multi-vehicle version of the NP-hard generalised travelling salesman problem.

As stated, the primary aim in the SBRP is to minimise the number of routes (buses) being used in a solution. It is therefore desirable to fill buses where possible, bringing parallels with the NP-hard bin-packing problem [6]. Indeed, if multistops were not permitted in a solution, then the identification of a solution using k routes while obeying Constraints (3)–(5) would result in a one-dimensional bin-packing problem with bin capacity m_c and item sizes equal to the number of students boarding at each bus stop. As noted, multistops *are* permitted in this SBRP meaning that students boarding at a particular bus stop can be assigned to different routes if needed (or, equivalently, items in the corresponding packing problem can be split across different bins). This allows us to produce a solution $\mathcal{R} = \{R_1, \dots, R_k\}$ satisfying constraints (3)–(5) that meets the lower bound of $k = \lceil (\sum_{i=1}^n s(v_i)) / m_c \rceil$, though of course these routes could be rather long.

From a different perspective, the issue of choosing the subset V_1' of bus stops to include in a solution is closely related to the set covering problem. Recall

that set covering involves taking a “universe” $U = \{1, 2, \dots, n\}$ and a set S whose elements are subsets of the universe, and seeks to find the smallest subset $S' \subseteq S$ whose union equals the universe. For example, given $U = \{1, 2, 3, 4\}$ and $S = \{\{1\}, \{1, 2\}, \{1, 3\}, \{3, 4\}, \{4\}\}$ the optimal solution is $S' = \{\{1, 2\}, \{3, 4\}\}$, containing just two elements.

Definition 2. $S' \subseteq S$ is a complete covering if and only if $\bigcup_{s \in S'} s = U$. A minimal covering is a complete covering in which the removal of any element in S' results in an incomplete covering.

According to Definition 2, an optimal solution to a set covering problem is a minimum cardinality solution among all minimal coverings. Note that while the task of identifying an optimal solution is NP-hard [6], the identification of minimal coverings is easily carried out in polynomial time. For example, starting with the complete covering $S' = S$, at each step we might simply remove any element $s \in S'$ for which $(S' - \{s\})$ is still a complete covering, repeating until S' is minimal.

With regards to the SBRP, using the bipartite graph $(V_1 - \{v_0\}, V_2, E_2)$, let S be the set whose elements correspond to the addresses within walking distance of each bus stop, $S = \{\Gamma(v) : v \in (V_1 - \{v_0\})\}$. According to Constraint (4), all addresses in a feasible solution must be served by a bus stop; hence, the task of identifying a subset $V_1' \subseteq V_1$ meeting this criterion is equivalent to the problem of finding a complete covering of the universe V_2 using the set S .

Theorem 2. Consider the SBRP in which multistops are not permitted (i.e., $R_i \cap R_j = \emptyset \forall R_i, R_j \in \mathcal{R}$), and let (V_1, E_1) be a graph whose pairwise distances satisfy the triangle inequality. Now let $\mathcal{R} = \{R_1, \dots, R_k\}$ be a solution satisfying Constraints (3)–(5) that has the minimum total journey time $\sum_{i=1}^k t(R_i)$. Then the subset of bus stops V_1' used in \mathcal{R} corresponds to a minimal covering S' .

Proof. The removal of any bus stop $v \in V_1'$ corresponds to the removal of the element $\Gamma(v)$ in S' which, by definition, results in an incomplete covering and violation of Constraint (4). Conversely, the addition of an extra element $\Gamma(v)$ to S' will result in a complete but non-minimal covering; however, this corresponds to the addition of an extra bus stop v in at least one route in \mathcal{R} which, due to the triangle inequality, will maintain or increase the total journey time of \mathcal{R} .

Note that this theorem does not hold when multistops are permitted in a solution. This is because the addition of an extra bus stop may allow a route to be shortened by redirecting it from a multistop and then through this new stop. The triangle inequality is also necessary, though it is acceptable here, being satisfied by both real-world road maps (where minimum distances/times between each pair of locations are used) and Euclidean graphs. Indeed, because of the extra delays incurred by dwell times in the SBRP, this inequality can be strengthened to $\forall u_1, u_2, u_3 \in V_1, t(u_1, u_2) + t(u_2, u_3) > t(u_1, u_3)$.

4 Algorithm Description

As noted, our strategy for this problem is to use a fixed number of routes k and allow the violation of Constraint (6) while ensuring that the remaining constraints (3)–(5) are always satisfied. Specialised operators are then used to try to shorten the routes in a solution such that Constraint (6) also becomes satisfied, giving a feasible solution. If this cannot be achieved at a certain computation limit, k is increased by one, and the algorithm is repeated. Initially, k is set to the lower bound $\lceil (\sum_{i=1}^n s(v_i)) / m_c \rceil$. An alternative approach would be to allow our search operators to alter k and then use its value as part of the objective function. However, evidence from the literature for similar partition-based problems suggests the former to usually be a more suitable approach [9, 13].

Our approach is based on iterated local search using a solution space \mathcal{V}'_1 that contains all bus stop subsets $V'_1 \subseteq V_1$ corresponding to minimal coverings. To begin, a member of \mathcal{V}'_1 is generated and improved via a local search routine (Section 4.2). Upon termination of this routine, a new member of \mathcal{V}'_1 is then generated by “kicking” the incumbent solution (Section 4.3), and re-running the local search. This is repeated until a stopping criterion is met (see Section 5).

4.1 Initial Solution and Cost Function

An initial solution $\mathcal{R} = \{R_1, \dots, R_k\}$ is constructed by first generating a subset of bus stops V'_1 corresponding to a minimal covering. In our case, this is achieved using the well-known greedy heuristic of Chvatal [3], followed by the removal of randomly selected bus stops (if necessary) until the covering is seen to be minimal.

Having generated V'_1 , the number of students boarding at each bus stop $s(v)$, $v \in V'_1$ is calculated. A variant of the first-fit descending heuristic for bin packing is then used to assign bus stops to routes. Specifically, at each step, the bus stop v with the largest number of boarding students is chosen and assigned to any route (vehicle) seen to have sufficient capacity. If no such route exists, then the route with the largest spare capacity x is chosen and v is assigned to this route along with x students. This has the effect of creating a multistop, since a copy of bus stop v , along with its remaining students will also need to be assigned to a different route in a subsequent iteration.

The above process produces a solution \mathcal{R} obeying Constraints (3)–(5). It is then evaluated according to an objective function $f(\mathcal{R}) = \sum_{i=1}^k t'(R)$, where

$$t'(R) = \begin{cases} t(R) & \text{if } t(R) \leq m_t \\ m_t + W(1 + t(R) - m_t) & \text{otherwise.} \end{cases} \quad (8)$$

Here, W introduces a penalty cost for routes whose journey times exceed the maximum m_t . In our case we set W to m_t and include the addition of one in the formula to ensure that a route with $t(R) > m_t$ is always penalised more heavily than two routes with individual journey times of less than m_t .

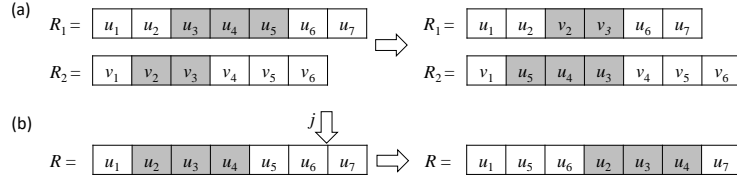


Fig. 2. Example application of (a) the Section Swap operator (here, the section u_3, u_4, u_5 has been inverted before insertion into R_2); and (b) the Extended Or-opt operator.

4.2 Local Search

As with many other VRP variants, our local search routine uses a combination of both inter- and intra-route neighbourhood operators. The following inter-route operators act on two routes $R_1, R_2 \in \mathcal{R}$. Without loss of generality, assume that $R_1 = (u_1, u_2, \dots, u_{l_1})$ and $R_2 = (v_1, v_2, \dots, v_{l_2})$.

Section Swap: Take two vertices in each route, u_{i_1}, u_{i_2} ($1 \leq i_1 \leq i_2 \leq l_1$) and v_{j_1}, v_{j_2} ($1 \leq j_1 \leq j_2 \leq l_2$), and use these to define the sections u_{i_1}, \dots, u_{i_2} and v_{j_1}, \dots, v_{j_2} within R_1 and R_2 respectively. Now swap the two sections, inverting either if this leads to a superior cost (see Fig. 2(a)).

Section Insert: Take a section in R_1 , defined by u_{i_1} and u_{i_2} as above, together with an insertion point j ($1 \leq j \leq l_2 + 1$) in R_2 . Now remove the section from R_1 and insert it before vertex v_j in R_2 , inverting the section if this leads to a better cost. If $j = l_2 + 1$, add the section to the end of R_2 .

Note that these inter-route operators may result in too many students being assigned to R_1 or R_2 , leading to a violation of Constraint (5). In our case, such moves are disallowed. Since multistops are permitted, it is also possible that they will result in a route containing a vertex $v \in V_1'$ more than once. Since each route must be a simple path, these need to be deleted. Assuming without loss of generality that a new route is to be produced by inserting the section u_{i_1}, \dots, u_{i_2} (possibly inverted) into a route $R = (v_1, v_2, \dots, v_{l_2})$, and that some vertex in the section is already present in R , this is done by simply removing the relevant vertex from the section and reassigning its students to the other occurrence of the vertex in R .

Our three intra-route neighbourhood operators act on a single route $R = (u_1, u_2, \dots, u_l) \in \mathcal{R}$. Their application does not affect the satisfaction of Constraints (3)–(5), nor do they introduce duplicate vertices into a route.

Swap: Take two vertices u_{i_1}, u_{i_2} ($1 \leq i_1 \leq i_2 \leq l$) in R and swap their positions.

2-opt: Take two vertices u_{i_1}, u_{i_2} ($1 \leq i_1 \leq i_2 \leq l$) and invert the section u_{i_1}, \dots, u_{i_2} within R .

Extended Or-opt: Take a section defined by u_{i_1} and u_{i_2} as above, together with an insertion point j outside of this section (i.e., $1 \leq j < i_1$ or $i_2 + 1 < j \leq l + 1$). Now remove the section and insert it before vertex u_j . If $j = l + 1$,

then add the section to the end of the route. Also, invert the section if this leads to a better cost (see Fig. 2(b)).

Note that, together, these five operators generalise a number of neighbourhood operators commonly featured in the literature. For example, our two inter-route operators include and extend the six outlined by Silva et al. [15] for the capacitated VRP. Similarly, they extend the basic VRP-based neighbourhood operators used with the bus routing problem considered in [14]. Our Extended Or-opt operator also generalises the more basic Or-opt, which only involves sections of up to three vertices [1].

Here, our local search procedure follows the steepest descent methodology: in each cycle all moves in all neighbourhoods are evaluated, and the move offering the largest reduction in cost is performed, breaking ties randomly. The process halts when no improving moves are identified. Note that the number of moves considered in each cycle is of $O(m^4)$, where $m = \sum_{i=1}^k |R_i|$ is the size of the incumbent solution. Though seemingly quite expensive, with appropriate book-keeping the changes in cost caused by individual applications of these neighbourhood operators can always be calculated in constant time. Consequently, this growth rate was not found to be particularly restrictive.

4.3 Generating New Minimal Coverings via a Kick Operator

While our local search routine is able to alter and improve the cost of a solution, it does not alter the subset of bus stops being used V'_1 . One way of doing this, as suggested in [14], would be to either swap a bus stop in a route with a currently unused bus stop, or simply remove a bus stop from a route altogether. However, besides not allowing the number of bus stops in V'_1 to increase, this is unsuitable here because it fails to ensure the satisfaction of Constraint (4).

Given a minimal subset of bus stops V'_1 , our operator first removes a randomly chosen non-compulsory bus stop $v \in V'_1$, followed by x further non-compulsory bus stops, leaving a partial covering.⁴ A different minimal covering is then constructed by selecting bus stops from the set $(V_1 - \{v\})$ using a randomised version of Chvatal's heuristic that, at each stage, arbitrarily selects any bus stop that will serve some currently unserved students, until all students are served. If necessary, randomly selected bus stops are then also removed until the covering is minimal.

Having produced a new minimal subset of bus stops $V''_1 \neq V'_1$, the current solution \mathcal{R} needs to be repaired to reflect these changes. To do this, the closest bus stops in V''_1 to each address are first recalculated and bus stops from the set $(V'_1 - V''_1)$ are deleted from routes in \mathcal{R} . Instances of multistops are also removed at this point so that each bus stop occurs at most once in \mathcal{R} . Randomly selected bus stops are then also removed from routes in \mathcal{R} if their number of students exceeds the maximum capacity m_c . Finally bus stops in V''_1 not yet in \mathcal{R} are

⁴ In our case a value for x is selected randomly according to a binomial distribution $X \sim B(|V'_1|, 3/|V'_1|)$.

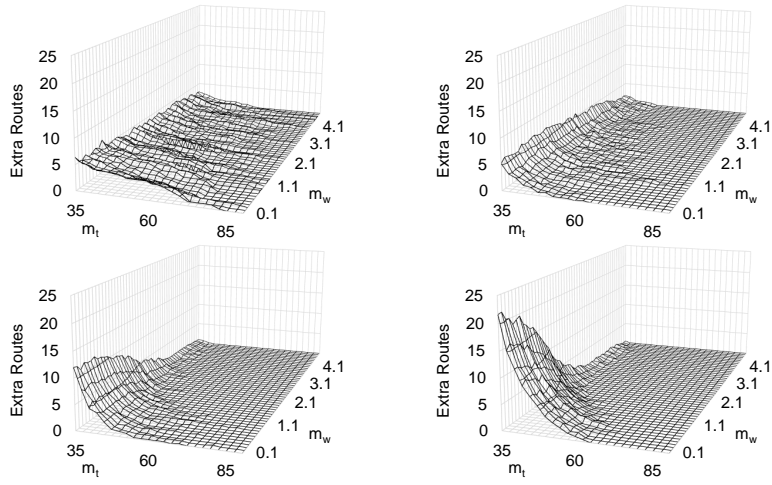


Fig. 3. Number of extra buses required by solutions for various values of m_t (in minutes) and m_w (in miles) for 25, 50, 100 and 250 bus stops respectively. Each point is the mean across five problem instances.

assigned to routes using the bin packing heuristic from Section 4.1. This results in a modified solution \mathcal{R} obeying Constraints (3)–(5) as desired.

5 Experimentation

Our experiments consider the issues that affect the number of extra buses required in a solution compared to the lower bound of $\lceil (\sum_{i=1}^n s(v_i)) / m_c \rceil$. To do this, artificial problem instances were generated by placing a school at the centre of a circle with radius $r > m_e$. Bus stops were then randomly placed within this circle, followed by a set of addresses, ensuring that each address was at least m_e distance units from the school, but within m_w distance units of at least one bus stop. Distances between vertices are assumed to be Euclidean.

Fig. 3 shows the effect of altering (a) the maximum walk distance m_w in our problem generator and (b) the maximum journey time m_t permitted by our algorithm, using 25, 50, 100 and 250 bus stops. In all instances we used a radius r of 15 miles and buses were assumed to travel at 30mph—hence all bus stops are within 30 minutes of the school. The number of addresses was set to 400, with the number of students boarding each bus stop $s(v)$ selected randomly from the set $\{1, 2, 3, 4\}$, giving approximately 1,000 students per-instance. Finally, the maximum bus capacity m_c and minimum eligibility distance m_e were set to 70 and 3 miles respectively, with ten seconds of execution time permitted for each value of k . (The algorithm was written in C++ and executed on a 3.3 GHz Windows 7 machine with 8 GB RAM.)

Fig. 3 demonstrates that more routes (buses) are needed when both the maximum journey times m_t and the maximum walking distances m_w are low.

For low values for m_t this is quite natural: shorter journey limits imply the need for more routes in feasible solutions. On the other hand, for low values of m_w the instance generator clusters addresses tightly around bus stops; consequently, nearly all bus stops are compulsory, making the problem very similar to that described in Theorem 1. This means that any savings that could be achieved by only using a subset of the bus stops are not available, creating a need for additional routes. We also see that these effects increase for larger numbers of bus stops where, for low values of m_w in particular, more bus stops will need to be visited.

Considering multistops, we found that these occur more frequently when it is advantageous or necessary to assign large numbers of students to individual bus stops. This occurs for high values of m_w , where students are able to walk larger distances to bus stops (implying fewer bus stops in V_1'), or when the number of bus stops is small. From a bin packing perspective, more students per-stop implies larger items to pack into the bins, meaning that more of these items will need to be “split”, resulting in a multistop.

As we might expect, the number of local optima visited by the algorithm within the ten second time limit (and therefore the number of kicks applied) is heavily influenced by the computational requirements of the local search routine, which is itself influenced by the size of a solution $\sum_{i=1}^k |R_i|$. To illustrate, for values of 25, 50, 100 and 250, these figures were seen to be approximately 250,000, 47,000, 4,000 and 60 respectively, suggesting that longer run times may be required for problem instances involving larger solutions.

6 Conclusions and Further Work

This paper has analysed a real-world school bus routing formulation that builds on previous models proposed in the literature by including bus stop selection, multistops, and dwell times. In doing so, relationships have been drawn with three well-known combinatorial optimisation problems.

Our experiments have demonstrated that our algorithm is often able to find solutions using the lower bound of $\lceil (\sum_{i=1}^n s(v_i)) / m_c \rceil$ routes. This is particularly so for instances where only a small proportion of bus stops need to be used, such as when the maximum walking distance of students is set quite high. Note, however, that in cases where all bus stops need to be used, our proposed kick operator has no effect, so it may be better to focus on extending the local search operator by, for example, including a tabu element.

As noted, the solution space in our current algorithm is restricted to bus stop subsets that correspond to minimal set coverings. However, according to Theorem 2, our use of multistops means that optimal solutions to a particular problem instance may not occur within this space. Future research will determine whether this restriction is beneficial, or whether it is preferable to use the larger space of all set coverings.

This paper has limited the empirical analysis to artificially generated problems; however, our research is ongoing and we are currently using this same

method with large real-world problems generated using web mapping services. One feature of our current solutions to these problems is that, by ensuring the set of used bus stops corresponds to a minimal covering, large numbers of students are often assigned to a relatively small number of bus stops, rather than using more convenient bus stops that are closer to their home. We expect further improvements to the service might therefore be achieved by sometimes allowing additional bus stops to be used, though perhaps without increasing the number of routes unduly.

References

1. G. Babin, S. Deneault, and G. Laporte. Improvements to the Or-opt heuristic for the symmetric travelling salesman problem. *Journal of the Operational Research Society*, 58(3):402–407, 2007.
2. R. Bertini and A. El-Geneidy. Modeling transit trip time using archived bus dispatch system data. *Journal of Transportation Engineering*, 130(1), 2004.
3. V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.
4. G. Dantzig and J. Ramser. The truck dispatching problem. *Management Science*, 60(1):80–91, 1959.
5. B. Eksioglu, V. Volkan, and A. Reisman. The vehicle routing problem: A taxonomic review. *Computers and Industrial Engineering*, 57(4):1472–1483, 2009.
6. M. Garey and D. Johnson. *Computers and Intractability - A guide to NP-completeness*. W. H. Freeman and Company, San Francisco, first edition, 1979.
7. G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43:408–416, 2009.
8. A. Letchford, J. Lysgaard, and R. Eglese. A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, 58:1642–1651, 2007.
9. R. Lewis. *A Guide to Graph Colouring: Algorithms and Applications*. Springer, 2015.
10. J. Park and B. Kim. The school bus routing problem: A review. *European Journal of Operational Research*, 202:311–319, 2010.
11. V. Pillac, M. Gendreau, C. Guéret, and A. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225:1–11, 2013.
12. Transit Cooperative Research Program. *Transit Capacity and Quality of Service Manual*, 3rd edition. isbn: 978-0-309-28344-1.
13. H. Qin, W. Ming, Z. Zhang, Y. Xie, and A. Lim. A tabu search algorithm for the multi-period inspector scheduling problem. *Computers and Operations Research*, 59:78–93, 2015.
14. P. Schittekat, J. Kinable, K. Sörensen, M. Sevaux, F. Spieksma, and J. Springael. A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, 229:518–528, 2013.
15. M. Silva, A. Subramanian, and L. Satoru Ochi. An iterated local search heuristic for the split delivery vehicle routing problem. *Computers and Operations Research*, 53:234–239, 2015.
16. C. Wang, Y. Zhirui, W. Yuan, X. Yueru, and W. Wei. Modeling bus dwell time and time lost serving stop in China. *Journal of Public Transportation*, 19(3):55–77, 2016.